

Treball de Fi de Grau

Grau en Enginyeria en Tecnologies Industrials

FamiSalut: Aplicació web que permet la gestió de cites mèdiques de tota la família

MEMÒRIA

Autor: Berta Alsina Hernández
Director: Lluís Solano
Convocatòria: Setembre 2019



Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona



Resum

Aquest projecte consisteix en el disseny i programació d'una aplicació web app amb l'objectiu de facilitar la gestió de la salut de l'usuari que l'utilitzi i dels seus familiars o persones al seu càrrec.

Aquest projecte s'ha dividit en tres fases principals. L'anàlisi del problema, el disseny de la solució i la implementació d'aquesta.

Es va iniciar el projecte definint clarament quin era el problema al que la web app volia aportar una solució i quina seria aquesta. El problema que es va detectar en el tractament de dades mèdiques va ser la falta d'unificació d'aquestes ja que es trobaven repartides en diferents plataformes. Per altra banda, es va detectar la facilitat de perdre el compte de quines cites mèdiques havia de realitzar cada familiar sobretot en èpoques de la vida on les revisions mèdiques son molt freqüents i importants (infància i vellesa). Un cop estudiades aquestes mancances es va plantejar com solució la web app FamiSalut.

Al llarg del treball s'ha dissenyat una aplicació web intuïtiva i fàcil d'utilitzar que permeti gestionar les cites mèdiques de tots els familiars i que permeti tenir tota la informació mèdica important de cada familiar en una mateixa plataforma.

Cal mencionar que s'ha considerat molt important el disseny gràfic de la web app ja que és un factor molt influent de cara al client. Es vol que la web app sigui senzilla i fàcil d'utilitzar per que qualsevol persona pugui fer-la servir.

Com a fase final s'ha realitzat la implementació de la web app el que ha implicat aprendre a programar amb diferents llenguatges de programació com son HTML, CSS i Python i aprendre a utilitzar diferents eines de programació com és el framework Flask. Al llarg de la fase d'implementació s'ha dut a terme una exploració de com portar una idea a la realitat i s'han anat resolent els problemes que han anat apareixent.

El resultat d'aquest projecte ha estat la realització d'un estudi de com solucionar una necessitat existent a través de la programació d'una aplicació web funcional i fàcil d'utilitzar.

Sumari

RESUM	3
SUMARI	4
LLISTAT DE FIGURES	6
LLISTAT DE TAULES	8
1. GLOSSARI	9
2. INTRODUCCIÓ	10
2.1. Motivació	10
2.2. Objectiu	11
2.2.1. Objectiu principal	11
2.2.2. Objectius específics	11
2.3. Abast del projecte	12
3. ANÀLISI DEL PROBLEMA	13
3.1. Descripció del problema.....	13
3.1.1. Funcionalitats.....	14
3.2. Aplicacions existents.....	14
3.2.1. La Meva Salut.....	14
3.2.2. ClepIO.....	15
4. DISSENY	17
4.1. Aplicació web	17
4.2. Estructura de FamiSalut.....	19
4.2.1. Interfície gràfica	19
4.2.2. Servidor	26
4.2.3. Estructura de dades.....	26
5. IMPLEMENTACIÓ	28
5.1. Desenvolupament	28
5.1.1. Tecnologies utilitzades	28
5.1.1.1. HTML	28

5.1.1.2. CSS	30
5.1.1.3. Flask	31
5.1.1.4. Python.....	34
5.2. Estructura dels scripts.....	35
5.3. Resolució de les funcionalitats.....	36
5.3.1. Calendari	36
5.3.2. Creació de familiars i metges	37
5.3.3. Visualització de la informació dels familiars i dels metges	39
5.3.4. Cerca	40
5.4. Flux entre pantalles.....	44
5.5. Tractament de dades	52
6. PROVES I AVALUACIÓ	54
7. PLANIFICACIÓ I COSTOS	55
7.1. Planificació temporal	55
7.1.1. Duració de les fases	56
7.2. Costos	57
6. TREBALL FUTUR	59
CONCLUSIONS	60
AGRAÏMENTS	61
BIBLIOGRAFIA	62
Referències bibliogràfiques	62
Bibliografia complementària	62

Llistat de figures

Figura 1: Logotip La Meva Salut.....	14
Figura 2: Logotip CleplO.....	15
Figura 3: Esquema de la comunicació entre client i servidor.	17
Figura 4: Estructura de la web app.....	18
Figura 5: Pàgina d'inici	20
Figura 6: Pàgina de cerca.....	20
Figura 7: Pàgina de cerca per familiar	21
Figura 8: Pàgina de cerca per familiar i per nom.....	21
Figura 9: Pàgina de cerca per familiar i per grup sanguini	21
Figura 10: Pàgina de resultat de la cerca per familiar	22
Figura 11: Pàgina de cerca per metge	22
Figura 12: Pàgina de cerca per metge i per nom	22
Figura 13: Pàgina de cerca per metge i per especialitat	23
Figura 14: Pàgina de resultat de la cerca per metge.....	23
Figura 15: Pàgina de Calendari	23
Figura 16: Pàgina de família.....	24
Figura 17: Pàgina afegir familiar.....	24
Figura 18: Pagina on es mostra la informació del familiar.....	24
Figura 19: Pàgina de metges.....	25
Figura 20: Pàgina afegir metge	25
Figura 21: Pàgina on es mostra la informació del metge	25
Figura 22: Logotip HTML5	28
Figura 23: Estructura d'un element HTML.....	29
Figura 24: Logotip CSS	30
Figura 25: Estructura d'una regla CSS	31
Figura 26: Logotip de Flask	31
Figura 27: Logotip de Python.....	34
Figura 28: Estructura de les carpetes de la web app	35
Figura 29: Google Calendar	36
Figura 30: Esquema del sistema de cerca	41
Figura 31: Pàgina d'Inici de FamiSalut.....	44
Figura 32: Pàgina de Cerca de FamiSalut.....	45

Figura 33: Pàgina Calendari de FamiSalut	45
Figura 34: Pàgina Família de FamiSalut.....	45
Figura 35: Pàgina Metges de FamiSalut.....	46
Figura 36: Pàgina de Cerca per Familiar	46
Figura 37: Pàgina de Cerca per Familiar i per Nom.....	47
Figura 38: Pàgina de Cerca per Familiar i per Grup Sanguini	47
Figura 39: Pàgina de Cerca per Metge	48
Figura 40: Pàgina de Cerca per Metge i per Nom	48
Figura 41: Pàgina de Cerca per Metge i per Especialitat.....	48
Figura 42: Pàgina on es mostren els resultats de la Cerca	49
Figura 43: Pàgina Informació d'un familiar.....	49
Figura 44: Pàgina afegir una cita	50
Figura 45: Formulari per afegir un nou Familiar.....	51
Figura 46: Formulari per afegir un nou Metge.....	51
Figura 47: Esquema del flux entre pantalles.....	52
Figura 48: Fitxer base de dades infofam.tx.....	52
Figura 49: Fitxer base de dades infomed.txt.....	53

Llistat de taules

Taula 1: Duració de les fases del projecte.....	56
Taula 2: Diagrama de Gantt	57
Taula 3: Costos totals del projecte	58

1. Glossari

Al llarg de la memòria d'aquest treball s'utilitzaran paraules tècniques o abreviatures. Aquestes estan explicades a continuació.

- **App:** Son les sigles que corresponen a la abreviatura de la paraula anglesa *Application* que vol dir aplicació.
- **CSS:** *Cascading Style Sheets* o Fulls d'Estil en Cascada és un llenguatge de disseny gràfic que s'utilitza per donar una bona presentació a documents escrits en llenguatge de marcat, defineix l'aspecte visual i el format de la pàgina web.
- **HTML:** *Hypertext Markup Language* o Llenguatge de Marcat d'Hipertext és un llenguatge de programació que s'utilitza per crear una web i amb el qual es defineix el contingut que es vol que tingui aquesta.
- **HTTP:** *HyperText Transfer Protocol* o Protocol de Transferència de Hipertext és un protocol de transferència de dades entre el client i el servidor. Ens permet accedir a qualsevol element d'internet.
- **MySQL:** és un sistema de gestió de dades basat en el llenguatge de consulta SQL.
- **Disseny web Responsive:** o adaptatiu. És un disseny web que permet la correcta visualització d'una pàgina web en qualsevol dispositiu ja sigui un ordinador, una tablet o un smartphone.
- **Servidor (web):** és un programa informàtic que segueix el protocol HTTP que serveix per processar una petició de part del client i realitza una resposta a aquestes sol·licitacions.
- **URL:** *Uniform Resource Locator* o Localitzador de Recursos Uniforme és una sèrie de caràcters que permet classificar/referenciar recursos en la Web de forma universal.
- **W3C:** *World Wide Web Consortium* o Consorci de la World Wide Web és un consorci internacional que defineix estàndards per la creació de pàgines web.
- **WWW:** *World Wide Web* o Xarxa Informàtica Global ens facilita l'accés a documents d'hipertext compartits via Internet.

2. Introducció

2.1. Motivació

Realitzar el disseny y la programació d'una web app com a treball de final de grau ve motivat pel meu interès en el món de la programació i de la informàtica.

En els últims anys la tecnologia ha avançat molt convertint-se en una eina indispensable per la nostra vida. Per tant, realitzar un treball on es dissenyi i es programi una web app em permetrà passar d'una idea teòrica a una realitat pràctica en l'àmbit de la tecnologia.

Fa uns anys, teníem tots els documents importants, ja fossin mèdics, acadèmics o professionals, en paper. Recordo com a casa, la meua mare, tenia una carpeta per cada fill amb tota la seva informació mèdica, acadèmica, etc. Allà emmagatzemava tota la informació important per trobar-la ràpida i fàcilment quan la necessités.

Aquest sistema va ser eficaç durant molts anys però ja fa un temps, que els informes mèdics, les receptes i els resultats de proves van començar a donar-se de forma electrònica. Per un costat aquest fet era una resposta a les necessitats d'un món on els avanços tecnològic eren cada cop més importants. Però per altra banda, aquest fet complicava la manera d'arxivar la informació mèdica de tota la família ja que la meitat dels informes es tenien en paper i l'altre meitat de manera electrònica.

Això em va fer pensar en com els avanços tecnològics han canviat la manera de fer les coses tant a nivell professional com a nivell particular. En una societat altament connectada on el temps es un valor molt important la salut devia ser-ho d'igual manera.

Després d'escoltar en moltes ocasions a la meua mare comentar lo bé que li aniria tenir tota la informació mèdica organitzada i disponible en tot moment vam tenir una idea. Aquesta idea consistia en crear una plataforma que anomenaríem FamiSalut, on cadascun pogués tenir la seva informació mèdica i la des seus familiars o persones al seu càrrec d'una manera ordenada i accessible.

Avui en dia existeixen aplicacions per fer quasi qualsevol cosa, perquè no facilitar aquesta tasca també amb una aplicació? La idea de FamiSalut consistia en desenvolupar una

aplicació que solucionés tant el problema d'emmagatzematge d'informació mèdica com el seguiment de cites mèdiques de tots els familiars. D'aquesta manera, aquesta plataforma permetria no només, tenir la informació mèdica de cada familiar d'una manera ordenada sinó que també permetria programar, controlar i visualitzar totes les cites mèdiques dels familiars en un calendari comú de manera que el control de totes les revisions fos àgil i fàcil.

Per tant, quan vaig haver de triar el tema del treball de final de grau no vaig dubtar en cap moment de què el volia fer. Volia crear una aplicació web que em permetria no només aprendre a programar amb HTML i CSS sinó que també em permetria desenvolupar l'idea teòrica que havia tingut a una realitat pràctica.

2.2. Objectiu

2.2.1. Objectiu principal

L'objectiu principal d'aquest treball es programar una aplicació web que permeti gestionar l'agenda de la salut i la informació mèdica més important d'una persona i dels seus familiars o persones que tingui al seu càrrec.

D'aquesta manera l'usuari principal podrà tenir aquesta informació en una mateixa plataforma on hi podrà accedir de manera intuïtiva i fàcil en el moment que sigui necessari i en qualsevol lloc del món.

També podrà programar, de forma fàcil i senzilla les seves cites mèdiques (tant de medicina preventiva com de seguiment) assegurant-se d'aquesta manera de que està realitzant totes les revisions mèdiques necessàries o prescrites pels diferents especialistes.

2.2.2. Objectius específics

Els objectius específics principals d'aquest projecte son els següents:

- Crear una aplicació web intuïtiva i fàcil d'utilitzar on es pugui:
 - o Gestionar totes les cites mèdiques de tots els familiars en un calendari comú.
 - o Gestionar una llista de familiars i una altra de metges.
 - o Emmagatzemar la informació mèdica important de cada familiar.
 - o Emmagatzemar la informació important de cada metge.

- Realitzar una cerca amb diferents filtres.
- Programar un full d'estil que encaixi perfectament amb la filosofia de la web app.

Pel que fa als objectius específics secundaris s'han plantejat els següents:

- Aprendre a programar amb HTML i CSS.
- Aprofundir en els meus coneixements de programació amb Python.
- Explorar quines dificultats o impediments apareixen al llarg del procés de realitzar un projecte en el qual es passa d'una idea teòrica a la programació de la mateixa.

2.3. Abast del projecte

Amb aquest projecte es pretén crear una web app per la gestió de la salut personal i familiar on l'usuari principal podrà:

- Programar visites mèdiques amb l'objectiu d'agilitzar el procés i el control d'aquestes.
- Visualitzar totes les cites mèdiques dels familiars en un calendari.
- Emmagatzemar la seva informació mèdica i la dels seus familiars creant així una llista de familiars i de metges amb la seva informació.
- Realitzar una cerca específica per metge o per familiar.

D'aquesta manera, aquesta plataforma facilitarà l'organització i l'emmagatzemament d'informació mèdica de tota una família.

El projecte està centrat en l'experiència del client a l'hora d'utilitzar l'aplicació web i per tant el treball s'ha centrat en el disseny i la programació d'una web app senzilla i fàcil d'utilitzar. En tot moment s'ha tingut en compte que aquesta web app seria una eina de gran ajut per l'organització familiar i per tant s'ha programat de manera que aquesta sigui *responsive* ja que això permet que l'usuari pugui accedir-hi des de qualsevol dispositiu electrònic.

Al llarg del projecte s'han fet proves de com diferents usuaris responien a l'aplicació web creada i un cop analitzades les respostes, es van fer les modificacions pertinents.

Pel que fa a la base de dades, com no ha sigut el punt de focus del treball, s'ha simulat amb un arxiu *.txt* on es guarda tota la informació afegida per l'usuari. Per tant, crear una base de dades adequada seria un àmbit de futur treball.

3. Anàlisi del problema

3.1. Descripció del problema

Al llarg de la nostra vida com pacients, acumulem una gran quantitat de documents mèdics. Ja siguin informes, anàlisis, resultats de proves o receptes.

Quan érem petits eren els pares els que s'encarregaven de recopilar tota la nostra informació mèdica i l'ordenaven de manera que fos fàcil trobar-la passat un temps. Però aquest sistema no era sempre efectiu ja que era molt fàcil perdre o arxivar de manera incorrecta algun documents i no es podia consultar en qualsevol moment (per consultar-ho un havia d'estar a casa).

Aquest sistema va ser encara menys efectiu en el moment que la nostra informació mèdica estava dividida entre la informació que teníem en mà i la informació digital. Però aquest fet a la vegada obria la porta a la possibilitat de gestionar les dades mèdiques de manera diferent. Per tant, la idea de FamiSalut neix al explorar aquestes necessitats que encara no tenen una eina per poder resoldre-les.

Quan els fills son petits, és molt important tenir en ment totes les revisions mèdiques que han de realitzar i si un no és molt organitzat és fàcil oblidar alguna revisió o alguna cita mèdica. El mateix passa quan un és fa responsable de la salut mèdica de persones grans. Donat que avui existeixen un munt d'aplicacions que ens fan més fàcil aquest tipus de tasques, en aquest treball es proposa la creació d'una web app que permeti organitzar la informació i les cites mèdiques de les persones al seu càrrec.

La solució que es proposa en aquest treball és crear una web app on es pugui tenir la informació mèdica important de cada familiar o persona al càrrec de l'usuari principal i on es puguin controlar les necessitats mèdiques de cadascun d'ells a través d'un calendari conjunt on apareixen totes les cites mèdiques de tota la família.

La motivació de crear una plataforma amb aquestes característiques neix de la necessitat dels pares i les mares de tenir a mà, en tot moment, la informació mèdica més important

dels familiars que estan al seu càrrec i de saber quan han de realitzar una visita mèdica, ja sigui per un tractament o per una revisió.

Aquesta web app, per tant, està enfocada a mares i pares que tenen fills petits o que es fan càrrec de les persones grans de la seva família.

3.1.1. Funcionalitats

Per solucionar el problema prèviament plantejat, s'ha dissenyat una web app que té les següents funcionalitats:

- Afegir familiars (Nom, cognom, data de naixement, sexe, grup sanguini i número de la targeta sanitària)
- Afegir metges (Nom, cognom, hospital on treballa, correu electrònic, telèfon, especialitat i familiars als quals visita)
- Buscador per:
 - o Familiar (per nom o per grup sanguini)
 - o Metge (per nom o per especialitat)
- Creació una cita en el calendari
- Visualització d'un calendari on apareguin les cites de tots els familiars.

3.2. Aplicacions existents

Existeix alguna aplicació semblant a la que es planteja per tant, a continuació s'exploraran les seves característiques.

3.2.1. La Meva Salut

“La Meva Salut és un espai digital segur que permet al ciutadà accedir a les seves dades personals de salut, fer consultes i realitzar tràmits electrònics.” [5]



Figura 1: Logotip La Meva Salut

La Meva Salut es una eina d'organització creada pel departament de salut de Catalunya que permet, d'entre altres coses:

- Accedir a la informació mèdica personal.
- Veure resultats de proves, exploracions, diagnòstics i informes clínics.
- Fer un seguiment de les vacunes administrades i per administrar.
- Trobar informació sobre les llistes d'espera dels centres.
- Comunicació entre pacient i professionals sanitaris.
- Realització de tràmits mèdics.

Per tant, es pot dir que La Meva Salut es una eina molt potent a l'hora de la gestió de la salut però només té en compte la sanitat pública. És a dir, resoluria tots els problemes de gestió de la gent que només és atesa per metges de la sanitat pública. En canvi, la gent que porta la seva salut per mútua o metges privats tindria, igualment, la seva informació mèdica separada en diferents plataformes.

La web app que es proposa, en canvi, no discriminarà cap tipus de sanitat i ajudarà a pacients tant de sanitat privada, pública o de les dues.

3.2.2. ClepIO

"ClepIO és l'eina en línia que et permet gestionar eficaçment la història de la teva salut i la dels teus." [4]



Figura 2: Logotip ClepIO

Aquesta aplicació està dotada de les següents funcionalitats:

- Tenir en una única plataforma tota la teva informació històrica sobre la teva salut.
- La possibilitat de donar accés a altres persones a la gestió de la teva informació. Tant a altres familiars com a metges.
- Seguiment eficaç d'infermetats cròniques com per exemple la diabetis.
- Configuració d'alertes quan es tingui una cita mèdica o quan s'hagi de prendre un medicament.

- Guardar documents mèdics escanejats.

El problema d'aquesta aplicació però, és la dificultat que apareix al voler utilitzar-la. No és intuïtiva ni està programada de cara a l'experiència de l'usuari.

Un dels objectius del treball per tant, serà crear una web app que compleixi les funcionalitats necessàries però que també sigui intuïtiva i fàcil d'utilitzar.

4. Disseny

4.1. Aplicació web

Abans de dissenyar la web app FamiSalut s'ha dut a terme un estudi de l'estructura general d'una aplicació web i de com aquesta realitza la comunicació entre l'usuari/client, el servidor i la base de dades.

Primer de tot cal definir el terme aplicació web o web app. Una web app es una aplicació informàtica que és utilitzada per un usuari que accedeix a un servidor web a través d'internet fent servir un navegador. Per tant hi ha una comunicació client servidor que té la següent estructura:



Figura 3: Esquema de la comunicació entre client i servidor.

El client realitza una petició a partir d'un URL per accedir al servidor. En el moment que l'usuari executa el URL es genera una petició HTTP des del costat client cap al servidor. El servidor l'executa i retorna una resposta al client en forma de document web que serà visualitzat per ell.

Cal mencionar que les web apps tenen una estructura multicapa generalment formada per les 3 capes següents:

- **Interfície gràfica:**

És la primera capa de l'aplicació web, s'utilitza directament per l'usuari i es executat pel servidor.

El llenguatge més utilitzat per implementar aquesta capa és HTML per tant és el llenguatge que s'utilitzarà per implementar aquesta capa a la web app FamiSalut. També s'utilitzaran altres llenguatges complementaris com seria CSS que servirà per donar estil.

- **Servidor:**

Aquesta capa es l'encarregada a rebre les peticions enviades per l'usuari, interpretar-les i realitzar una resposta. El servidor també es l'encarregat de establir contacte amb la base de dades tant sigui per consultar-la o per actualitzar-la.

Hi ha moltes tecnologies que es poden utilitzar per programar el servidor. En aquest cas s'ha utilitzat Flask que és un micro framework escrit amb el llenguatge de programació Python. S'ha escollit aquest llenguatge ja que és un llenguatge amb el que s'havia treballat prèviament.

- **Base de dades:**

La base de dades és on s'emmagatzema tota la informació important per la funcionalitat de la nostra aplicació web. Hi ha molts tipus de bases de dades però en aquest treball, com s'ha centrat l'atenció en les capes de la interfície gràfica i del servidor s'ha simulat una base de dades amb un fitxer .txt que és llegit a través d'una funció Python. Per tant, d'ara en endavant, parlarem d'una estructura de dades en comptes d'una base de dades ja que és el que s'ha programat en aquest projecte.

Per tant, l'estructura completa de la web app és la següent:

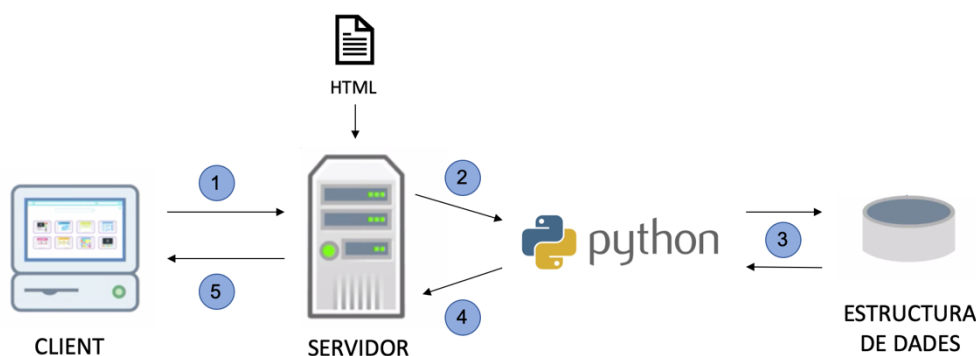


Figura 4: Estructura de la web app.

1. L'usuari interactua amb el servidor realitzant una petició a través del seu navegador web.
2. El servidor rep la informació i executa les funcions Python que té emmagatzemades.
3. S'obtenen dades de l'estructura de dades i aquesta es modifica si s'escau.
4. El servidor recull la informació processada per les funcions.
5. El servidor envia els resultats al client a través d'internet.

4.2. Estructura de FamiSalut

La web app que es dissenyarà al llarg del treball estarà composta de les 3 capes prèviament mencionades. A continuació s'especificarà el disseny escollit per cadascuna d'elles.

4.2.1. Interfície gràfica

Un cop establertes les funcionalitats que es volia que tingués FamiSalut es va plantejar un disseny gràfic de la web app que fos senzill, intuïtiu i fàcil d'utilitzar.

Avui en dia no només es ven un producte per la tecnologia que hi ha darrera sinó que el disseny forma una part molt important de l'èxit que tingui aquest producte quan surti al mercat. Un exemple d'això és l'èxit que han tingut els telèfons iPhone envers a altres telèfons que duen a terme les mateixes funcionalitats.

Es per això que el disseny gràfic de la web app programada s'ha considerat un aspecte molt important d'aquesta i per tant, durant el treball, s'explorarà la relació entre la funcionalitat i el disseny de la web app. Per fer això, en aquesta capa de la web app s'han utilitzat els llenguatges de programació: HTML i CSS.

A l'hora de començar a dissenyar la web app programada es va fer un anàlisi de les pantalles necessàries per dur a terme totes les funcionalitats. Es va escollir tenir un índex a la part esquerra de la pantalla que permetés accedir a totes les pàgines principals (Inici, Cerca, Calendari, Família i Metges) des de qualsevol pantalla de la web app.

Per tant, es va partir de cinc pàgines principals i un cop aquestes van ser dissenyades, es van afegir dotze pantalles més que permetien poder dur a terme totes les funcionalitats de la web app.

A continuació es pot veure el disseny que es va realitzar inicialment per cada pantalla de la web app. Cal mencionar que les pantalles que es mostren a continuació són un prototipus del que es volia que fos la web app un cop programada.

1. Inici:



Figura 5: Pàgina d'inici

2. Cerca:



Figura 6: Pàgina de cerca

2.1. Cerca per familiar:



The screenshot shows the FamiSalut application interface. On the left is a vertical menu with icons and labels: 'Inicio' (home), 'Búsqueda' (search), 'Calendario' (calendar), 'Familia' (family), and 'Médicos' (doctors). The main area on the right has the 'FamiSalut' logo at the top. Below the logo, it says 'Buscar por:' followed by two blue buttons: 'Nombre' (Name) and 'Grupo sanguíneo' (Blood group).

Figura 7: Pàgina de cerca per familiar

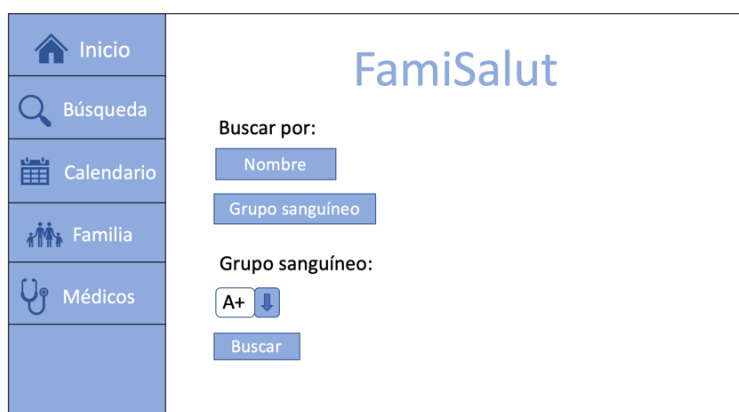
2.1.1. Cerca per familiar i per nom:



This screenshot is similar to Figure 7 but includes an additional search option. It shows the same menu and 'FamiSalut' logo. Under 'Buscar por:', there are buttons for 'Nombre' and 'Grupo sanguíneo'. Below these, there is a section labeled 'Nombre:' with a text input field and a blue 'Buscar' (Search) button.

Figura 8: Pàgina de cerca per familiar i per nom

2.1.2. Cerca per familiar i per grup sanguini:



This screenshot shows the search interface for blood groups. It includes the same menu and 'FamiSalut' logo. Under 'Buscar por:', there are buttons for 'Nombre' and 'Grupo sanguíneo'. Below these, there is a section labeled 'Grupo sanguíneo:' with a dropdown menu showing 'A+' and a blue 'Buscar' (Search) button.

Figura 9: Pàgina de cerca per familiar i per grup sanguini

2.1.3. Resultat de la cerca per familiar:

<div>Inicio</div> <div>Búsqueda</div> <div>Calendario</div> <div>Familia</div> <div>Médicos</div>	<div>FamiSalut</div> <div>Resultado de la búsqueda:</div> <div>Familiar 1 <div>Info</div></div> <div>Familiar 2 <div>Info</div></div> <div>Familiar 3 <div>Info</div></div>
---	---

Figura 10: Pàgina de resultat de la cerca per familiar

2.2. Cerca per metge:

<div>Inicio</div> <div>Búsqueda</div> <div>Calendario</div> <div>Familia</div> <div>Médicos</div>	<div>FamiSalut</div> <div>Buscar por:</div> <div><div>Nombre</div></div> <div><div>Especialidad</div></div>
---	---

Figura 11: Pàgina de cerca per metge

2.2.1. Cerca per metge i per nom:

<div>Inicio</div> <div>Búsqueda</div> <div>Calendario</div> <div>Familia</div> <div>Médicos</div>	<div>FamiSalut</div> <div>Buscar por:</div> <div><div>Nombre</div></div> <div><div>Especialidad</div></div> <div>Nombre:</div> <div><div></div></div> <div><div>Buscar</div></div>
---	--

Figura 12: Pàgina de cerca per metge i per nom

2.2.2. Cerca per metge i per especialitat:

The screenshot shows the FamiSalut search interface. On the left is a vertical sidebar with navigation links: Inicio (home icon), Búsqueda (magnifying glass icon), Calendario (calendar icon), Familia (family icon), and Médicos (stethoscope icon). The main content area has the FamiSalut logo at the top. Below it, there's a 'Buscar por:' section with two buttons: 'Nombre' and 'Especialidad'. Underneath is an 'Especialidad:' section with a dropdown menu currently showing 'Otorrino' and a 'Buscar' button.

Figura 13: Pàgina de cerca per metge i per especialitat

2.2.3. Resultat de la cerca per metge

The screenshot shows the search results page. The sidebar is identical to the previous page. The main content area displays the FamiSalut logo and the heading 'Resultado de la búsqueda:'. Below this, there are three entries: 'Médico 1', 'Médico 2', and 'Médico 3', each followed by an 'Info' button.

Figura 14: Pàgina de resultat de la cerca per metge

3. Calendari:

The screenshot shows the calendar page. The sidebar is identical to the previous pages. The main content area features the FamiSalut logo and a calendar widget for July 2019. The calendar shows dates from 1 to 31, with the 10th and 11th highlighted in yellow. To the right of the calendar is a 'Nueva cita' button. At the bottom of the calendar widget, there is a small note: 'Los eventos se muestran en la zona horaria: Hora de Europa central - Madrid'.

Figura 15: Pàgina de Calendari

4. Família:



FamiSalut

Añadir familiar

Familiares:

Familiar 1 [Info](#)

Familiar 2 [Info](#)

Familiar 3 [Info](#)

Figura 16: Pàgina de família

4.1. Crear familiar:



FamiSalut

Nombre:

Apellido:

Fecha de nacimiento (dd/mm/aaaa):

Sexo:

☒ Mujer

☐ Hombre

☐ Otro

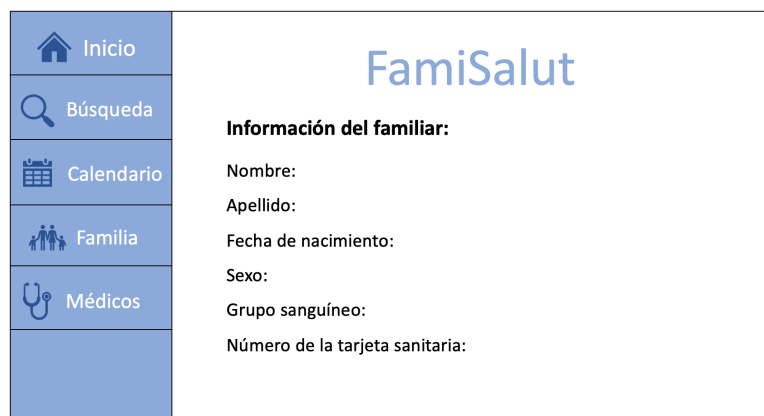
Grupo sanguíneo:

Número de la tarjeta sanitaria:

[Crear](#)

Figura 17: Pàgina afegir familiar

4.2. Informació del familiar:



FamiSalut

Información del familiar:

Nombre:

Apellido:

Fecha de nacimiento:

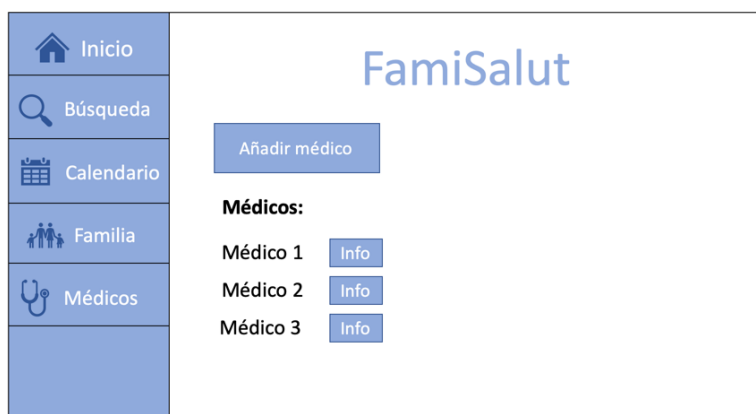
Sexo:

Grupo sanguíneo:

Número de la tarjeta sanitaria:

Figura 18: Pagina on es mostra la informació del familiar

5. Metges



FamiSalut

Añadir médico

Médicos:

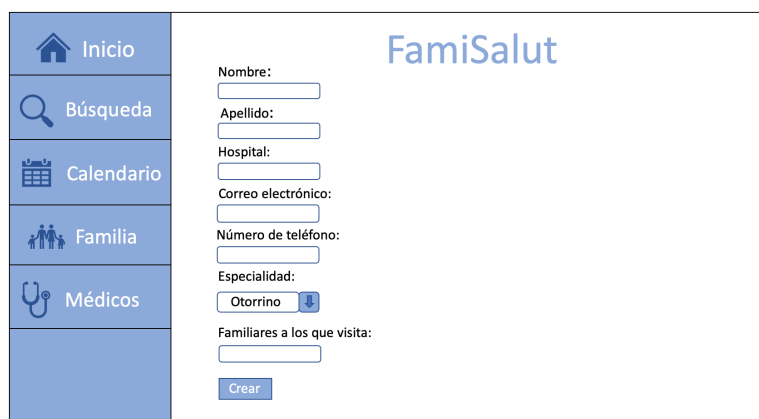
Médico 1 [Info](#)

Médico 2 [Info](#)

Médico 3 [Info](#)

Figura 19: Pàgina de metges

5.1. Crear metge:



FamiSalut

Nombre:

Apellido:

Hospital:

Correo electrónico:

Número de teléfono:

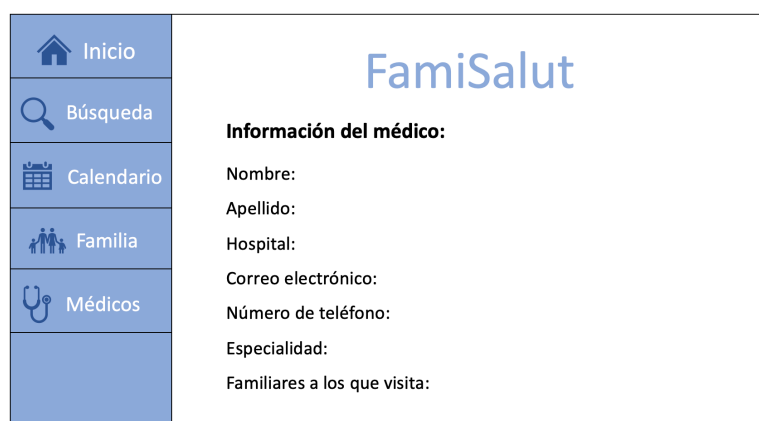
Especialidad:

Familiares a los que visita:

[Crear](#)

Figura 20: Pàgina afegir metge

5.2. Informació del metge:



FamiSalut

Información del médico:

Nombre:

Apellido:

Hospital:

Correo electrónico:

Número de teléfono:

Especialidad:

Familiares a los que visita:

Figura 21: Pàgina on es mostra la informació del metge

Com s'ha mencionat prèviament, aquest disseny és el que es reproduirà un cop la web app sigui creada. És per això que s'ha programat un full d'estil amb CSS des de zero que s'adapti perfectament al disseny gràfic plantejat.

4.2.2. Servidor

La funció del servidor és ser capaç de respondre a totes les peticions que se li envien de part del client. És a dir, és el que permet que es duguin a terme les funcionalitats plantejades i per tant és una de les parts més importants de la programació de la web app.

En aquest projecte, s'ha escollit treballar amb Flask, un micro framework escrit amb el llenguatge de programació Python, per dur a terme la programació del servidor de la web app.

Depenent del llenguatge que s'utilitzi per programar el servidor aquest s'estructurarà d'una manera diferent. En el cas de la web app FamiSalut, com s'ha escollit treballar amb el llenguatge Python, s'utilitzaran diferents funcions que atenguin a cada tipus de petició.

4.2.3. Estructura de dades

Pel que fa a la base de dades, el primer que es va dur a terme va ser escollir quines dades s'emmagatzemarien, és a dir, quines dades es demanarien de cada familiar i de cada metge.

En el cas dels familiars es va escollir demanar la següent informació:

- Nom
- Cognom
- Data de naixement
- Sexe
- Grup sanguini
- Número e targeta sanitària

Es va considerar que amb aquesta informació es podia descriure a un familiar perfectament. El nom, cognom, data de naixement i sexe eren dades necessàries per descriure'l com a persona i el grup sanguini i el número de targeta sanitària es van

considerar dues informacions mèdiques bàsiques necessàries per qualsevol emergència mèdica.

En canvi, en el cas dels metges es va escollir demanar la informació que es mostra a continuació:

- Nom
- Cognom
- Hospital on treballa
- Correu electrònic
- Número de telèfon
- Especialitat
- Familiars als que visita

En aquest cas, es va considerar que com a dades generals només es necessitava el nom i el cognom del metge però com a dades mèdiques específiques, era necessari emmagatzemar en quin hospital treballava, quin era el seu correu electrònic i el seu número de telèfon per si un es volia posar amb contacte amb ell, quina era la seva especialitat mèdica i per últim, a quins familiars visitava.

Un cop escollida la informació que es demanaria de cada familiar i de cada metge es va procedir a escollir la base de dades en la que s'emmagatzemaria aquesta informació.

En comptes de crear una base de dades amb algun sistema de gestió de dades com seria MySQL es va decidir simular una estructura de dades amb un parell de fitxers .txt que emmagatzemarien tota la informació necessària d'una manera ordenada. Per tant, la web app plantejada no tindrà una base de dades sinó una estructura de dades.

5. Implementació

5.1. Desenvolupament

Un cop analitzat el problema i dissenyada la solució s'ha procedit a implementar-la. A continuació s'explicaran totes les tecnologies utilitzades per la creació de la web app FamiSalut i l'estructura final d'aquesta.

5.1.1. Tecnologies utilitzades

Pel desenvolupament de la web app s'han utilitzat diferents llenguatges de programació que s'explicaran detalladament a continuació:

5.1.1.1. HTML

Les sigles HTML corresponen a: Llenguatge de Marcat d'Hipertext. Aquest llenguatge és el que s'utilitza per crear una web i amb el qual es defineix el contingut que es vol que tingui aquesta.



Figura 22: Logotip HTML5

El llenguatge HTML consta d'etiquetes i de contingut. Les etiquetes descriuen el contingut de manera que el navegador web entengui l'estructura de la pàgina.

Com es veurà al llarg de la implementació de la web app, els documents HTML estan compostats de diferents elements. S'anomena element al conjunt de: etiqueta d'inici, contingut i etiqueta final

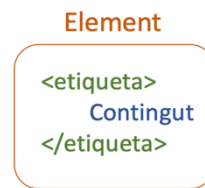


Figura 23: Estructura d'un element HTML

A continuació es mencionaran els tipus d'etiquetes més utilitzats al llarg d'aquest treball i s'explicarà per que serveixen:

- `<h>`: S'utilitza per indicar que el text que hi ha a continuació és un títol.
- `<p>`: S'utilitza per definir un paràgraf de text. Per tant, aquest estarà separat de la resta d'elements de la pàgina per espais buits.
- `
`: Al utilitzar-lo s'insereix un espai buit. Equival a fer clic al botó de enter i per tant permet deixar espais entre paràgrafs o altres tipus d'elements.
- `<a>`: S'utilitza per definir un enllaç. Per tant es un text que et permet accedir a un altre URL al clicar-lo.
- `<div>`: És un element genèric que s'utilitza per replegar un tros de codi del document. S'utilitza per organitzar la disposició d'elements a la pantalla quan al mateix document es fan servir diferents tipus de continguts.
- `<input>`: Especifica un camp on el client pot introduir-hi dades per tan es un element interactiu. Hi ha diferents tipus de inputs com serien: text, *button* o *checkbox*.
- `<form>`: Crea un formulari per que sigui omplert pel client. També és un element interactiu.
- ``: S'utilitza per iniciar una llista on cadascun dels elements esta definit per etiquetes ``.

L'estructura general d'una pàgina web creada amb HTML està dividida en dos parts principals. El *head* que correspon al encapçalat y el *body* que és el cos del programa.

Al *head* s'inclouen les metadades que són dades del document HTML. Les metadades normalment defineixen l'idioma del codi i realitzen el link amb els fulls d'estil que s'utilitzaran per donar forma a la pàgina web. És també al *head* on s'ha especificat que la web app sigui *responsive*.

El *body* està format pels elements que formen part de la pàgina web. És en aquesta part on s'han utilitzat totes les etiquetes i elements prèviament mencionats.

5.1.1.2. CSS

Utilitzarem el llenguatge CSS per personalitzar l'aplicació web. Com s'ha mencionat en l'apartat anterior, els documents HTML defineixen el contingut que es vol que aparegui en la nostra web app. Amb CSS el que es farà és dissenyar la presentació d'aquesta.

Les sigles CSS corresponen a *Cascading Sytle Sheets* que vol dir: Fulls d'Estil en Cascada. CSS és un llenguatge de disseny gràfic que s'utilitza per donar una bona presentació a documents escrits en llenguatge de marcat (HTML).



Figura 24: Logotip CSS

Una de les avantatges de crear un full d'estil separat dels documents HTML és que permet crear una web amb un format i un estil homogeni.

És al full d'estil on s'especifica: color, mida, tipus de lletra i altres aspectes visuals importants per que l'experiència de l'usuari a l'hora d'utilitzar la web app.

Com s'ha mencionat prèviament, un dels objectius del treball és crear una interfície senzilla i fàcil d'utilitzar. Per tant, s'ha considerat molt important la creació d'un full d'estil CSS que compleixi aquets objectius.

Però com està estructurat un full d'estil CSS? Aquest consisteix en una sèrie de regles i aquestes regles tenen l'estructura següent:

Estan formades per un selector i una propietat.



Figura 25: Estructura d'una regla CSS

El selector es la part de la regla que especifica a quina etiqueta va referida. En l'exemple de la figura 25, la regla fa referència a l'etiqueta `<h1>` i per tant tot el text que formi part del contingut de `<h1>` s'escriurà amb la font Helvètica. La propietat, en canvi, especifica l'estil gràfic que seguirà l'etiqueta seleccionada.

5.1.1.3. Flask

Flask es un micro framework escrit amb Python creat per facilitar el desenvolupament de web apps. Per tant, és una eina que ens permet crear una aplicació web amb Python.



Figura 26: Logotip de Flask

Flask ha sigut creat per abordar les tasques més comuns del desenvolupament de web apps i s'enfoca en fer-ho d'una manera senzilla i minimalista. Per tant, és el framework adequat per crear un prototipus ràpid de qualsevol projecte.

Abans de continuar, cal definir que és un framework. "Un framework és una eina que ens proporciona un esquema de treball i un seguit d'utilitats i funcions que ens facilita la creació de pàgines webs dinàmiques" [9]

Un cop definit el terme framework cal especificar com s'ha introduït Flask dins la nostra web app.

Com s'ha comentat prèviament, es vol que, quan el client faci una petició, el servidor sigui capaç d'executar-la i respondre adequadament. Per tant, al llarg d'aquest projecte s'han programat funcions que respondran a cada petició que es faci pel client/usuari.

Aquestes funcions respondran a les peticions de tres maneres diferents.

- Retornaran un fitxer HTML per que l'usuari pugui visualitzar-lo.
- Realitzaran un operació interior com seria realitzar una cerca.
- Realitzaran el contacte amb la base de dades ja sigui per llegir-la com per afegir-hi informació.

Per entendre com funciona el servidor de la web app cal entendre l'estructura d'aquestes funcions i per tant, aquesta s'explicarà a continuació.

Pel que fa a les funcions que retornen un fitxer HTML totes tenen l'estructura següent:

```
@app.route('/busqueda')
def bus():
    return
    render_template('busqueda.html')
```

Es defineix una funció, en aquest cas `bus`, que retorna el fitxer *busqueda.html*. Hi ha una variació d'aquest tipus de funcions quan no només es vol retornar un fitxer HTML sinó que també es volen recollir dades d'un formulari omplert pel client. En aquest cas, les funcions programades tenen l'estructura següent:


```
@app.route('/formufam', methods=["POST"])
def formufam():
    nombre = request.form["nombre"]
    apellido = request.form["apellido"]
    fecha = request.form["fecha"]
    sexo = request.form["gender"]
    grupsang = request.form["sangre"]
    tarjsan = request.form["tarjsan"]
    persona=[nombre, apellido, fecha, sexo, grupsang, tarjsan]
    creapers(persona)
    return render_template('formufam.html')
```

Pel que fa a les funcions que simplement realitzen una operació estan estructurades de la següent manera:

```
def busquedafn(info, fam):
    resposta=[]
    for pers in fam:
        persona=fam[pers]
        if persona[0]==info:
            resposta.append(persona)
    return resposta
```

Aquesta funció compara una informació obtinguda amb aquella informació de cada familiar per veure si hi ha una coincidència i retorna una llista de llistes on cada llista es una persona de la família.

Per últim, pel que fa a les funcions que realitzen un contacte amb la base de dades, s'ha de tenir en compte que llegeixen o escriuen dins d'un fitxer i això ho fan com s'indica a continuació:

```
def llegirfitxer1(nomf):  
    n = 'app/scripts/'+ "infofam.txt"  
    d={}  
    with open(n,'r') as f:  
        for linea in f:  
            linea=linea.strip()  
            l=linea.split(",")  
            d[str(l[0])]=l  
    return (d)
```

Inicialment s'obre el fitxer seleccionat que es troba al directori app/scripts/, es llegeix i es guarda la informació obtinguda dintre d'un diccionari.

5.1.1.4. Python

Python és un llenguatge de programació molt potent i sovint utilitzat per la programació de web apps degut a la seva gran versatilitat.



Figura 27: Logotip de Python

Com s'ha mencionat prèviament, el micro framework Flask està escrit amb Python i per tant en us llenguatge que s'ha hagut de dominar per poder realitzar la programació del servidor de la web app.

5.2. Estructura dels scripts

La carpeta que engloba tot el contingut necessari per que la web app funcioni està estructurat de la següent manera ja que hem utilitzat Flask.

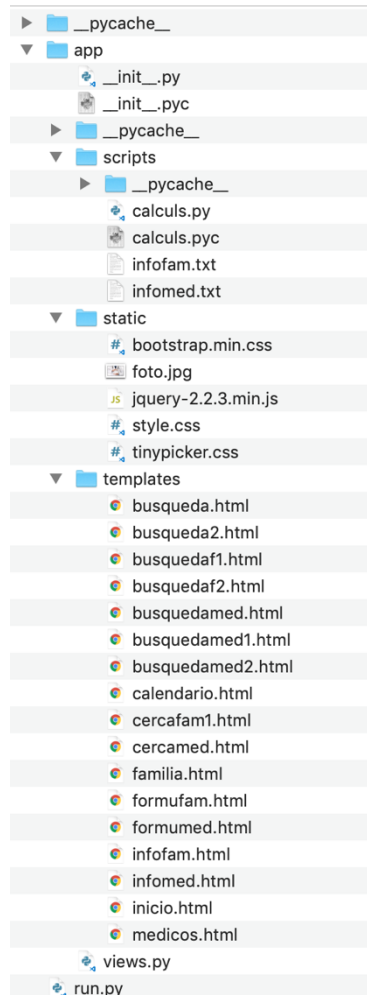


Figura 28: Estructura de les carpetes de la web app

Les carpetes importants pel desenvolupament de la web app son les carpetes: templates, static i __pycache__ que es troben dintre de la carpeta app. La carpeta templates consta de tots els documents HTML que formen la web app, la carpeta static és on es troba tot el que va referit a l'estil de la web app i per últim, a la carpeta __pycache__ trobem els scripts python que realitzen els càlculs necessaris per dur a terme les funcionalitats de la web app com seria per exemple tractar les dades que es troben als fitxers .txt.

5.3. Resolució de les funcionalitats

Hi ha una sèrie de funcions/accions programades que permeten que la web app dugui a terme les funcionalitats que se li ha establert. A continuació s'analitzaran cada una d'elles.

5.3.1. Calendari

La funcionalitat més important que havia de tenir FamiSalut és la visualització d'un calendari on es poden veure i afegir totes les cites mèdiques de tots els familiars.

La visualització del calendari i la capacitat d'afegir una cita mèdica en aquest van de la mà per tant, a continuació, s'explicarà amb detall com s'ha aconseguit dur a terme aquestes funcionalitats.

A l'hora de dissenyar el calendari, s'ha escollit un calendari ja existent on es podrien afegir les cites mèdiques d'una manera ràpida i fàcil. S'ha escollit utilitzar Google Calendar ja que és un calendari a la disposició de tothom, per tant, pel que fa a les funcionalitats: d'afegir cites mèdiques i visualitzar de totes elles en un calendari conjunt s'ha decidit utilitzar el calendari següent:

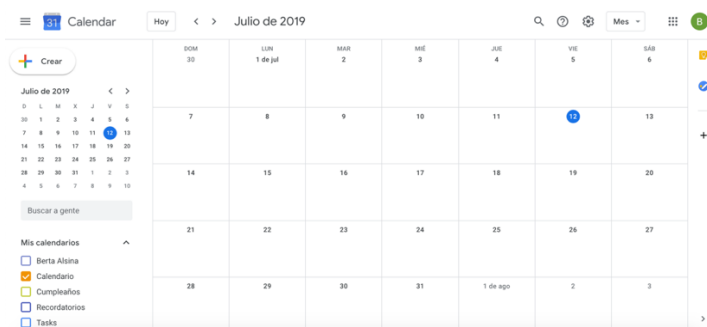


Figura 29: Google Calendar

Per encastar aquest calendari a la web app, s'ha utilitzat l'etiqueta <iframe> ja que aquesta es fa servir per encastar un document a l'HTML actual.

Aquesta funció es du a terme en el fitxer calendario.html i té aquesta estructura:

```
<iframe  
src="https://calendar.google.com/calendar/embed?src=kpar4mrut3fi7mpm5bg  
ie7evg8%40group.calendar.google.com&ctz=Europe%2FMadrid" style="border:  
0" width="800" height="600" frameborder="0" scrolling="no"></iframe>
```

A continuació es va afegir un botó que connectés la pantalla de l'agenda de FamiSalut amb la pantalla de creació d'una cita dintre el calendari Google Calendar. Això s'ha fet utilitzant l'etiqueta `<a>` creant així un hyperlink amb la creació de Google Calendar.

Aquesta funció també es du a terme en el fitxer `calendar.html` i té aquesta estructura:

```
<a href="https://calendar.google.com/calendar/r/eventedit"  
target="_blank" id="calendario" class="button">Añadir cita</a>
```

5.3.2. Creació de familiars i metges

Per afegir un familiar o un metge a l'estructura de dades de la web app es segueix el següent procediment.

Primer, es recull tota la informació d'un familiar o d'un metge obtinguda a partir del formulari omplert pel client. A continuació es pot veure com es recull aquesta informació.

```
@app.route('/formufam', methods=["POST"])
def formufam():
    nombre = request.form["nombre"]
    apellido = request.form["apellido"]
    fecha = request.form["fecha"]
    sexo = request.form["gender"]
    grupsang = request.form["sangre"]
    tarjsan = request.form["tarjsan"]
    persona=[nombre, apellido, fecha, sexo, grupsang, tarjsan]
    creapers(persona)
    return render_template('formufam.html')
```

Un cop rebuda la informació, aquesta s'assigna a una variable com és: nombre, apellido o gruposang. A continuació, es crea una llista anomenada persona amb tota la informació d'aquesta ordenada sempre de la mateixa manera : persona=[nombre, apellido, fecha, sexo, gruposang, tarjsan]. Per últim, és crida a la funció creapers i se li proveeix la variable persona.

La funció creapers es troba en el fitxer càlculs.py i té aquesta estructura.

```
def creapers(persona):
    f = open ('app/scripts/infofam.txt','a')
    for e in persona:
        f.write(str(e)+",")
    f.write(" \n")
    f.close()
```

Aquesta funció, el que fa es obrir el fitxer on es troba tota la informació de tots els familiars i l'obre de tal manera que pot afegir informació al fitxer. A continuació recorre la llista proveïda amb la informació del familiar en qüestió i l'afegeix a la base de dades del fitxer infofam.txt.

Per crear un metge es fa de la mateixa manera però en comptes de cridar a la funció `creapers()` es crida a la funció `creamed()` també programada al fitxer `calculs.py`.

5.3.3. Visualització de la informació dels familiars i dels metges

Un cop s'ha creat un familiar o un metge i s'ha guardat la seva informació a l'estructura de dades és important tenir-la ben ordenada (com s'exposa al capítol 5.5. Tractament de dades) per quan es vulgui consultar.

Una altra de les funcionalitats importants de FamiSalut es poder consultar la informació del metges i dels familiars (figura 45 i 43) ja sigui des de la pantalla de metges (figura 35) o la pantalla de familiars (figura 34) com des de l'eina de cerca (figures 39 i 36). Això s'ha aconseguit a partir de les següents funcions:

Primer de tot es va crear un arxiu HTML que mostrés per pantalla la informació del familiar o metge corresponent.

```
<p1> Informació del familiar <p1>
<br>
<br>
<p4>Nombre: {{info[0]|safe}} </p4>
<br>
<br>
<p4>Apellido: {{info[1]|safe}} </p4>
<br>
<br>
<p4>Fecha de nacimiento: {{info[2]|safe}} </p4>
<br>
<br>
<p4>Sexo: {{info[3]|safe}} </p4>
<br>
<br>
<p4>Grupo Sangíneo: {{info[4]|safe}} </p4>
<br>
<br>
<p4>Número de Tarjeta Sanitaria: {{info[5]|safe}} </p4>
```

Com es pot apreciar en aquest tros de codi, la informació mostrada per pantalla es dinàmica i depèn del familiar o metge del que es vol consultar la informació. Això ha implicat una dificultat afegida a l'hora de programar-ho però finalment s'ha resolt creant un botó que té com a valor el nom de la persona de la qual es vol consultar la informació.

```
{%for e in gent%}  
    <form action="/mostrainfofam", method="post">  
    <p4>{{e}}</p4>  
    <!-- Creo un boton con el id=nombre del familiar -->  
    <button name="info" type="submit" value={{e}}>Info</button>  
    </form>  
    <br>  
    <br>  
{% endfor %}
```

D'aquesta manera, quan es clica aquest botó, s'enviaria una informació al servidor com si aquest fos un formulari i amb aquesta informació es pot cridar al document infofam.html mostrat anteriorment.

```
@app.route('/mostrainfofam', methods=["POST"])  
def mostrainfofam():  
    global fam  
    nom=request.form["info"]  
    print(nom)  
    info3=fam[nom]  
    return render_template('infofam.html', info=info3)
```

5.3.4. Cerca

Per últim, és important mencionar com s'han dut a terme un seguit de funcions que permeten que dins la web app es pugui realitzar una cerca tant per familiar com per metge.

Primer de tot l'usuari ha d'escollir que és el que vol buscar d'entre les diferents opcions disponibles.

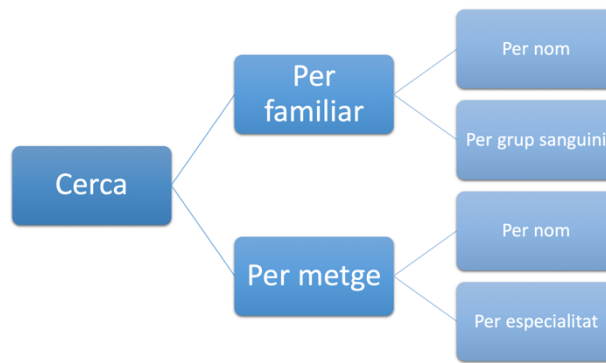


Figura 30: Esquema del sistema de cerca

Aquesta elecció la du a terme l'usuari clicant sobre els botons que especifiquen aquests filtres. Aquests botons s'han programat de la següent manera:

```
<a href="busqueda2" id="buscafam" class="button">Familiar</a>
```

Un cop arribat al filtre final apareix un formulari on s'haurà d'indicar el nom, el grup sanguini o l'especialitat que es vol cercar. Aquest formulari té una de les tres formes que es mostren a continuació.

Per nom:

```
<form action="/cercafam1" method="post">
  <p4>Nombre:</p4>
  <br>
  <br>
  <input type="text" name="nombref">
  <br>
  <br>
  <input type="submit" id="busqueda"
value="Buscar">
</form>
```

Per grup sanguini:

```
<form action="/cercafam2" method="POST">
  <p4>Grupo Sangíneo:</p4>
  <br>
  <br>
  <select id="sangre" name="sangre">
    <option value="apos">A+</option>
    <option value="aneg">A-</option>
    <option value="bpos">B+</option>
    <option value="bneg">B-</option>
    <option value="abpos">AB+</option>
    <option value="abneg">AB-</option>
    <option value="opos">O+</option>
    <option value="oneg">O-</option>
  </select>
  <br>
  <br>
  <input type="submit" id="busqueda" value="Buscar">
</form>
```

Per especialitat:

```
<form action="/ceramed2" method="post">
  <p4>Especialidad:</p4>
  <br>
  <br>
  <select id="especialidad" name="especialidad">
    <option value="otorr">Otorrino</option>
    <option value="ocul">Oculista</option>
    <option value="derma">Dermatólogo</option>
    <option value="pediatra">Pediatra</option>
    <option value="gine">Ginecologo</option>
    <option value="general">Medicina general</option>
    <option value="fisio">Fisioterapeuta</option>
    <option value="otro">Otro</option>
  </select>
  <br>
  <br>
  <input type="submit" id="busqueda" value="Buscar">
</form>
```

Un cop omplerts aquests formularis es recull la informació obtinguda de la manera que es mostra a continuació.

```
@app.route('/ceramed1', methods=["POST"])
def cercamed1():
    bus=request.form["nombre"]
    fam, med = llegir ()
    resultats=busquedamedn(bus, med)
    print (resultats)
    return render_template('ceramed.html', info=resultats)
```

Finalment, es crida a la funció busquedamedn(), busquedamede(), busquedafn() o busquedafs() depenent del que escaigui. Aquestes funcions busquen per metge i per nom,

per metge i per especialitat, per familiar i per nom i per familiar i per grup sanguini i tenen la següent estructura:

```
def busquedamedn (info, med):  
    resposta=[]  
    for medi in med:  
        medico=med[medi]  
        if medico[0]==info:  
            resposta.append(medico)
```

5.4. Flux entre pantalles

Un cop explicades les funcionalitats que tindrà la web app dissenyada cal especificar quin serà el flux entre pantalles.

Quan s'executi l'aplicació s'apareixerà a la pàgina d'inici i des d'aquesta es podrà accedir a les pàgines següents: Cerca, Agenda, Família i Metges a partir de l'índex lateral. A continuació es mostren les cinc pantalles principals de la web app. Cal mencionar que es pot accedir a aquestes pàgines des de qualsevol pantalla de l'aplicació.



Figura 31: Pàgina d'Inici de FamiSalut

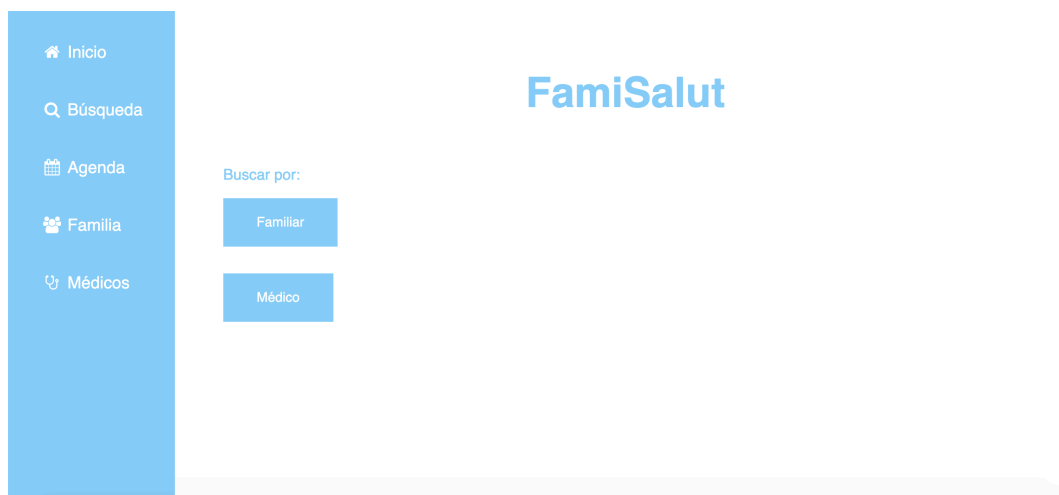


Figura 32: Pàgina de Cerca de FamiSalut

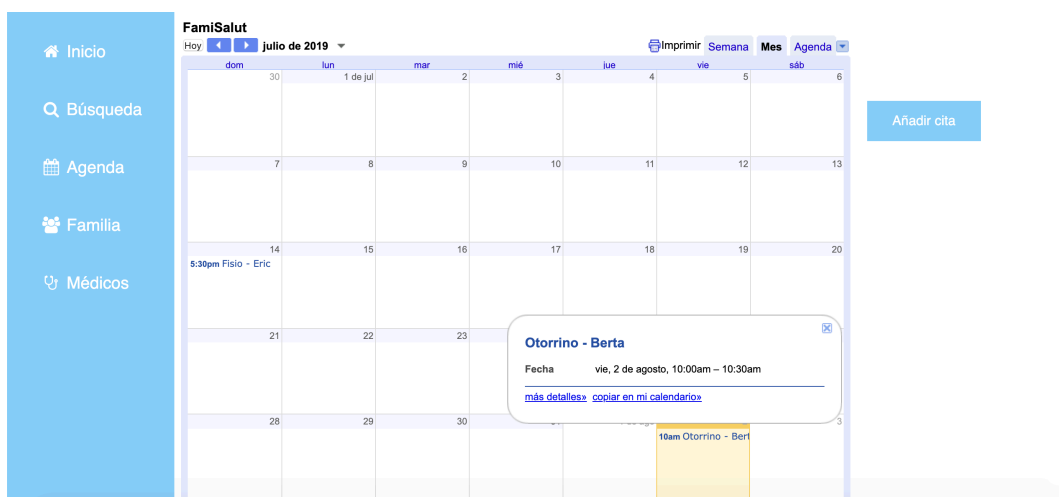


Figura 33: Pàgina Calendari de FamiSalut

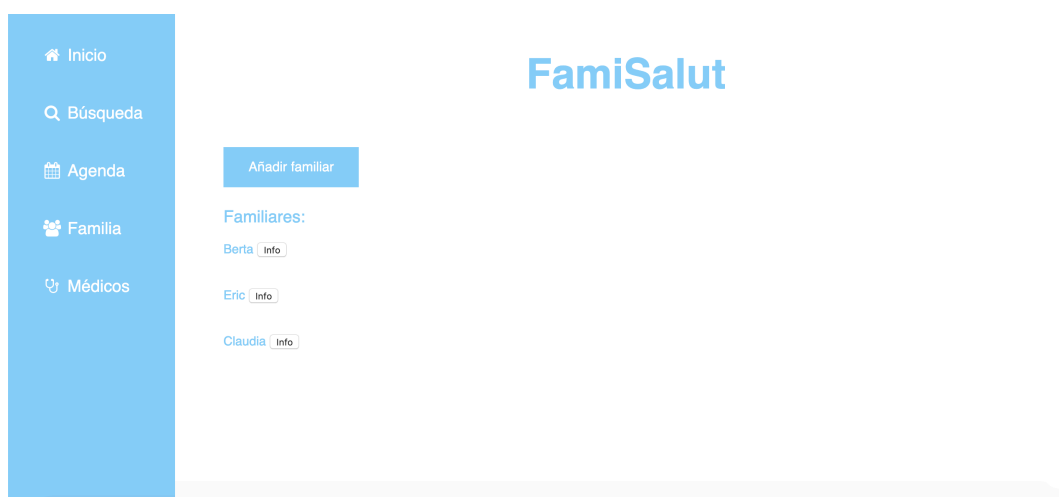


Figura 34: Pàgina Família de FamiSalut

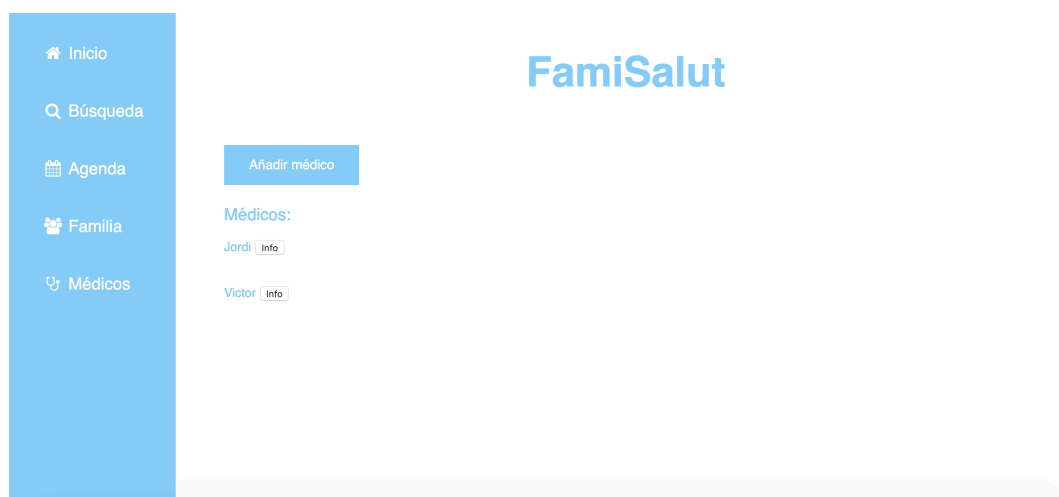


Figura 35: Pàgina Metges de FamiSalut

Com es pot veure a la figura 32 un cop s'entra en la pantalla de cerca es pot cercar per familiar o per metge. Un cop escollit el primer filtre de la cerca es pot escollir el segon filtre: per nom, per grup sanguini o per especialitat depenent de si es cerca per metge o per familiar. Per últim, un cop realitzada la cerca s'obtenen els resultats d'aquesta i a continuació, si es vol, es pot visualitzar la informació dels metges o familiars trobats.

A continuació es mostren les pantalles de cerca en l'ordre mencionat.

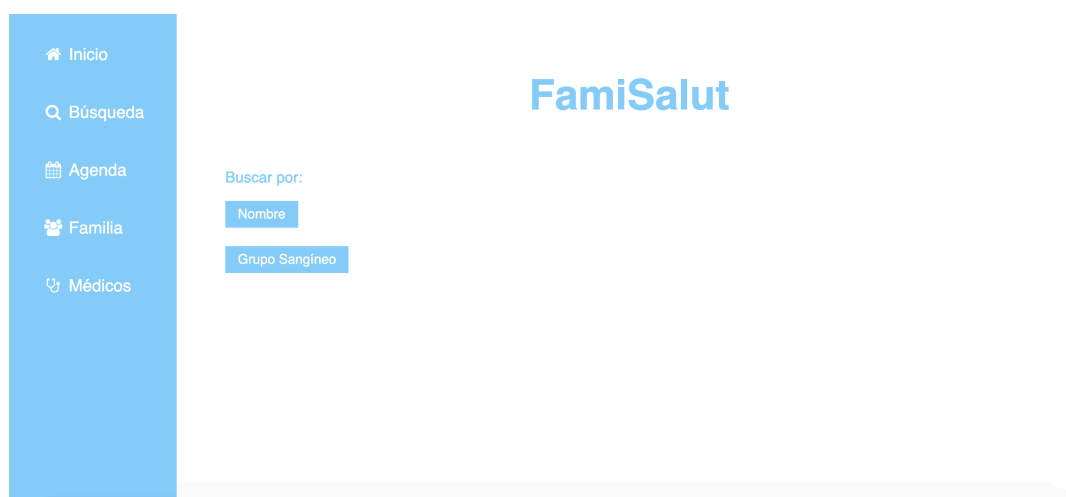


Figura 36: Pàgina de Cerca per Familiar

The screenshot shows the FamiSalut web application interface. On the left is a blue sidebar with navigation links: Inicio, Búsqueda, Agenda, Familia, and Médicos. The main content area has the 'FamiSalut' logo at the top. Below the logo, there is a search section titled 'Buscar por:'. It contains two blue buttons: 'Nombre' and 'Grupo Sanguíneo'. The 'Nombre' button is selected. Below these buttons, there is a text input field labeled 'Nombre:' and a blue 'Buscar' button.

Figura 37: Pàgina de Cerca per Familiar i per Nom

This screenshot shows the same FamiSalut web application interface as Figure 37, but with the 'Grupo Sanguíneo' button selected in the 'Buscar por:' section. The text input field below is now labeled 'Grupo Sanguíneo:' and contains the text 'A+'. The 'Buscar' button remains at the bottom.

Figura 38: Pàgina de Cerca per Familiar i per Grup Sanguini

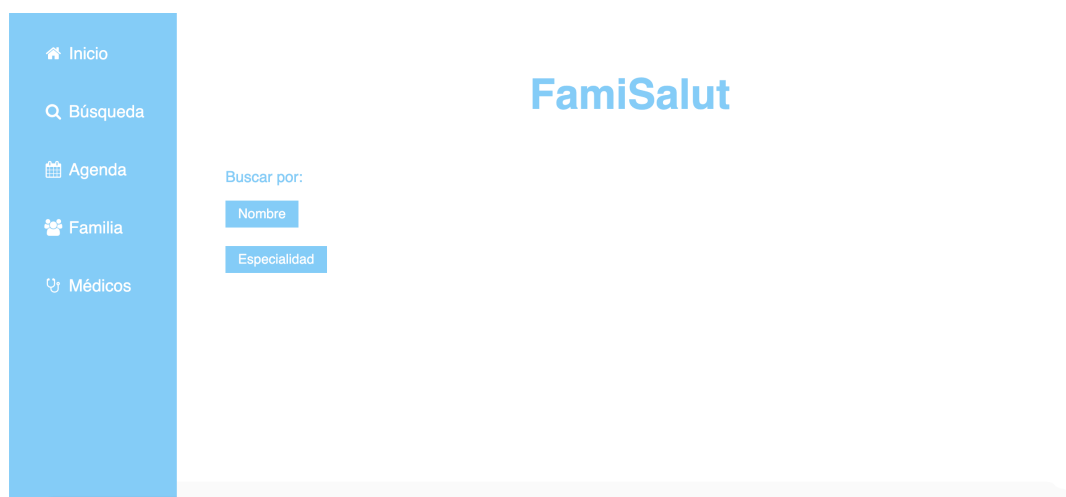


Figura 39: Pàgina de Cerca per Metge

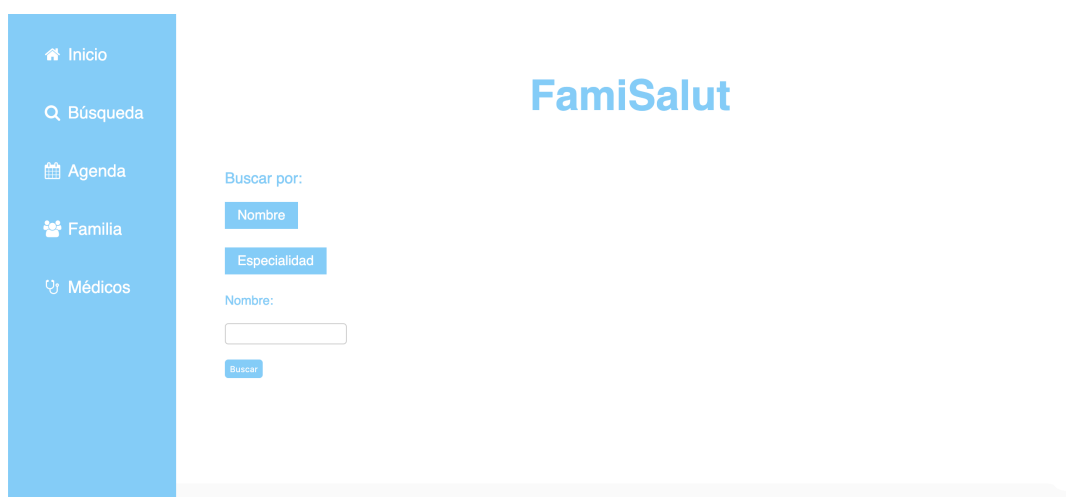


Figura 40: Pàgina de Cerca per Metge i per Nom

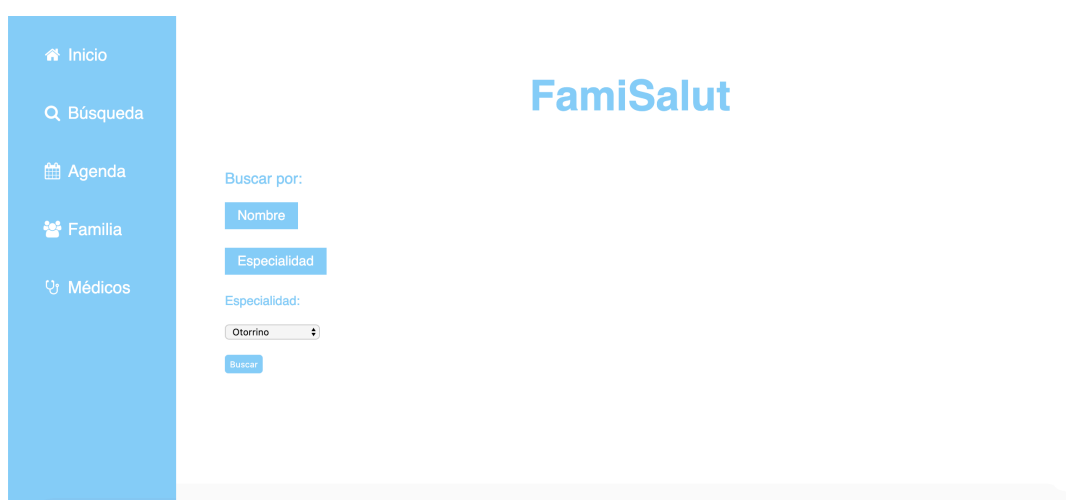


Figura 41: Pàgina de Cerca per Metge i per Especialitat



Figura 42: Pàgina on es mostren els resultats de la cerca per familiar



Figura 43: Pàgina Informació d'un familiar



Figura 44: Pàgina on es mostren els resultats de la cerca per metge



Figura 45: Pàgina Informació d'un metge

Pel que fa a l'agenda, des de la seva pantalla es pot veure el calendari amb totes les cites mèdiques de tots els familiars (com es pot veure a la figura 33) i es pot afegir una cita nova a aquest calendari gràcies al botó "Añadir cita". Quan es fa clic sobre el botó: afegir una cita, apareix la pantalla següent.

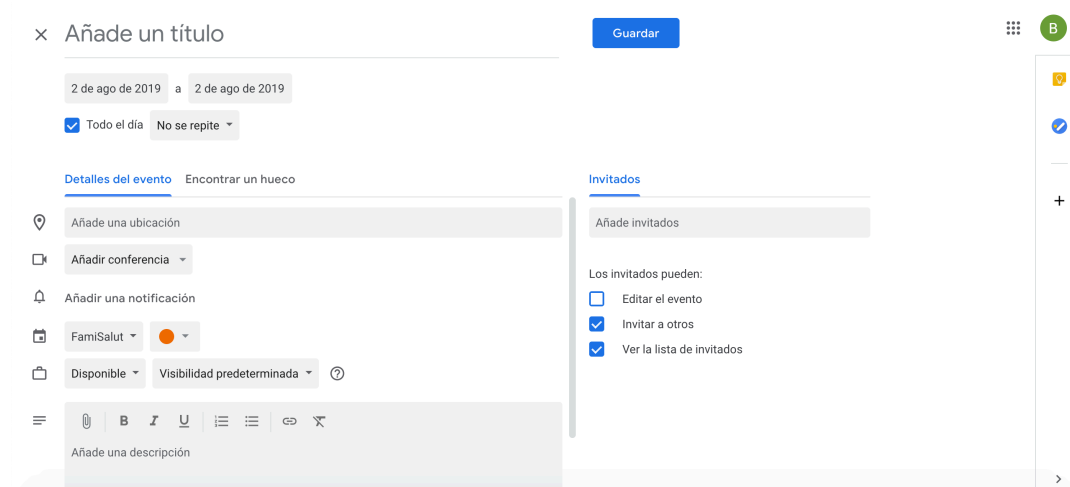
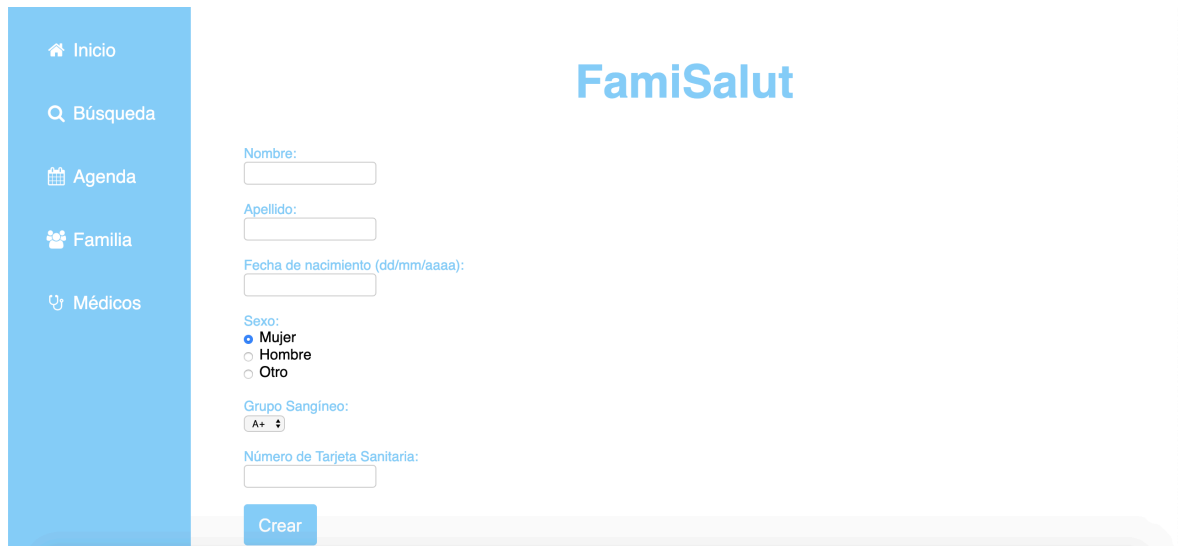


Figura 46: Pàgina afegir una cita

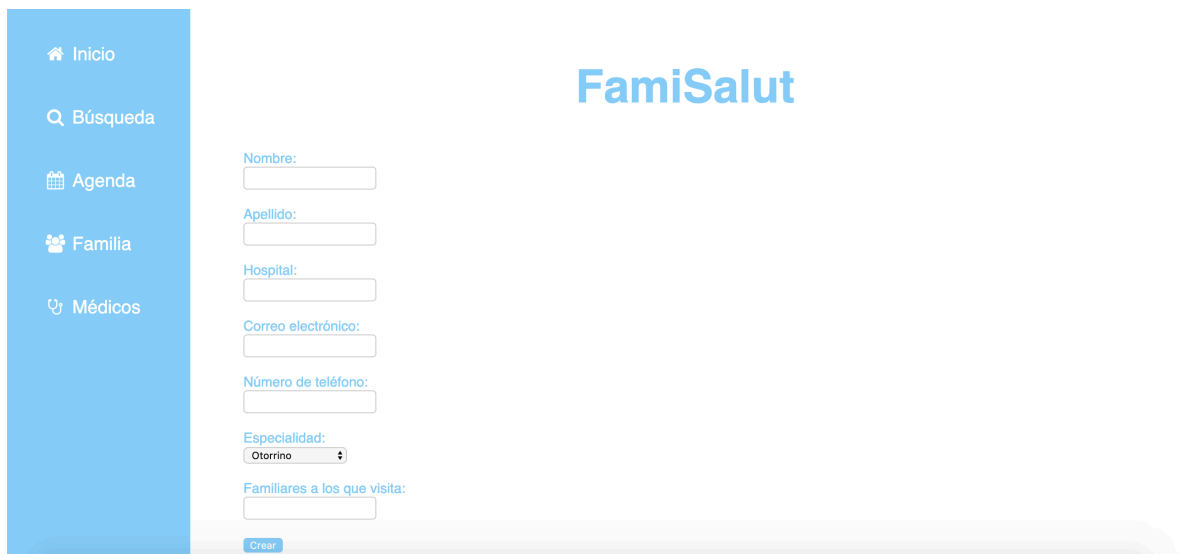
Per últim, les pantalles de Família i Metges estan estructurades de la mateixa manera. Es veuen tots els familiars/metges que ja existeixen podent fàcilment visualitzar la seva informació (figura 43 i 45 respectivament) i es pot afegir un nou familiar/metge fent clic sobre el botó Afegir Familiar/Metge (figura 47 i 48 respectivament).

Quan es vol afegir un nou familiar o un nou metge, apareixen els següents formularis que s'han d'omplir amb la informació necessària.



The screenshot shows the FamiSalut app interface. On the left is a blue sidebar menu with icons and text for 'Inicio', 'Búsqueda', 'Agenda', 'Familia', and 'Médicos'. The main area has the 'FamiSalut' logo at the top. Below it is a form with the following fields: 'Nombre:' (text input), 'Apellido:' (text input), 'Fecha de nacimiento (dd/mm/aaaa):' (text input), 'Sexo:' with radio buttons for 'Mujer' (selected), 'Hombre', and 'Otro', 'Grupo Sangüneo:' with a dropdown menu showing 'A+', and 'Número de Tarjeta Sanitaria:' (text input). At the bottom of the form is a blue 'Crear' button.

Figura 47: Formulari per afegir un nou Familiar



The screenshot shows the FamiSalut app interface, similar to the previous one. The sidebar menu is on the left. The main area has the 'FamiSalut' logo. The form for adding a doctor includes the following fields: 'Nombre:' (text input), 'Apellido:' (text input), 'Hospital:' (text input), 'Correo electrónico:' (text input), 'Número de teléfono:' (text input), 'Especialidad:' with a dropdown menu showing 'Otorrino', and 'Familiares a los que visita:' (text input). A blue 'Crear' button is at the bottom of the form.

Figura 48: Formulari per afegir un nou Metge

La següent figura mostra un esquema del flux entre pantalles de la web app FamiSalut que recopila tota la informació prèviament mencionada.

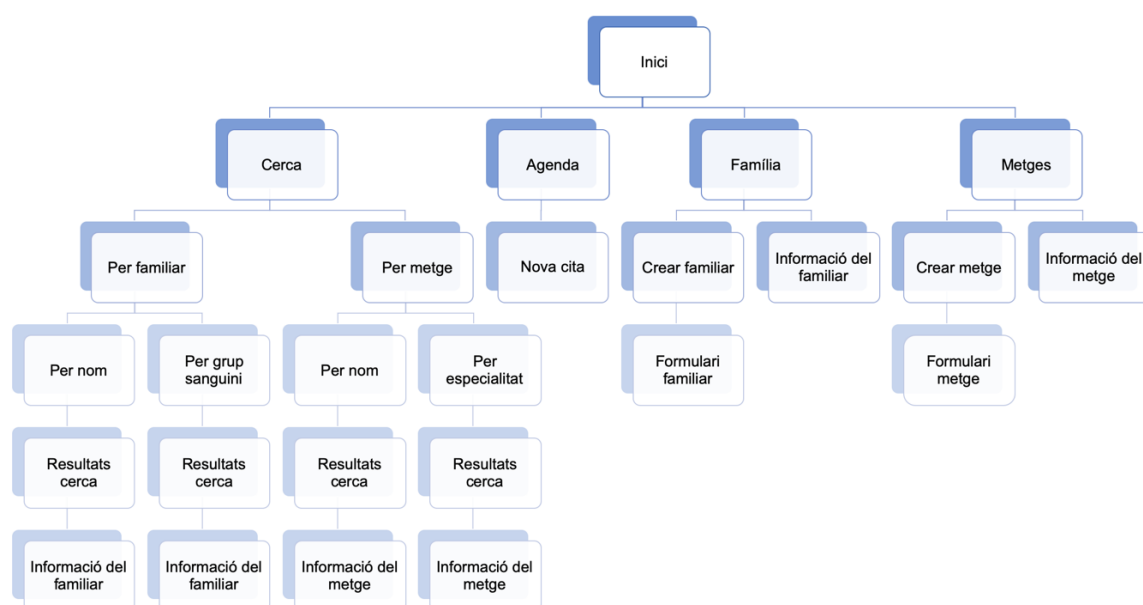


Figura 49: Esquema del flux entre pantalles

5.5. Tractament de dades

Com s'ha mencionat anteriorment, la web app programada no compta amb una base de dades sinó amb una estructura de dades creada per emmagatzemar tota la informació important de cada familiar o metge.

Aquests fitxers .txt s'han estructurat de la següent manera:

- Estructura de dades amb la informació dels familiars:

S'emmagatzema tota la informació d'un familiar en una línia on es separa per punts i coma la següent informació en aquest ordre: Nom, Cognom, Data de naixement, Sexe, Grup sanguini i Número de la targeta sanitària.

```

≡ infam.txt ✕
app ▸ scripts ▸ ≡ infam.txt
1   Berta;Alsina;04/10/1996;mujer;0+;45085949458;
2   Eric;Alsina;22/06/1998;hombre;A+;45085949458;
3   Claudia;Alsina;09/02/2001;mujer;A+;45085949458;
  
```

Figura 50: Fitxer base de dades infam.tx

- Estructura de dades amb la informació del metges:

Com en el cas del familiars, s'emmagatzema tota la informació d'un metge en una línia. La informació emmagatzemada és la següent: Nom, Cognom, Hospital on treballa, Correu electrònic, Número de telèfon, Especialitat i Familiars als que visita.

```
≡ infomed.txt x
app ▸ scripts ▸ ≡ infomed.txt
1   Jordi;Vazquez;Teknon;jordivazquez@hotmail.com;932800936;Fisioterapeuta; Eric i Berta;
2   Victor;Boj;Quiron;victorboj@quiron.com;934022931;Oculista; Claudia;
```

Figura 51: Fitxer base de dades infomed.txt

6. Proves i avaluació

Al llarg de la programació de l'aplicació web s'ha comprovat el seu correcte funcionament, s'ha donat a provar a persones externes al projecte i s'han realitzat les modificacions necessàries tenint en compte aquests dos inputs.

La programació de l'aplicació web s'ha dut a terme de la següent manera i sempre s'ha compaginat programació de nous elements amb detecció d'errors de tota la web app.

El que es va fer inicialment va ser programar "l'esquelet" de la web app, és a dir, es van dissenyar i programar totes les pantalles de la web app amb tots els detalls, inclosos els botons, però no se li va donar cap funcionalitat ni un lligam amb la base de dades (en aquesta fase només es va utilitzar HTML i CSS).

Un cop fet això es va confirmar que amb aquestes pantalles es poguessin dur a terme totes les funcionalitats plantejades a l'inici del projecte i es van fer canvis per facilitar tant la usabilitat de la web app com la facilitat de programar aquestes funcionalitats.

A continuació es va dur a terme la programació més funcional de la web app on es va utilitzar Python i la base de dades a part de HTML i CSS.

Aquesta va ser la fase més complicada del projecte i la que va requerir més temps. Un cop acabada aquesta fase es va fer servir l'aplicació i es van detectar alguns errors com per exemple que cada cop que s'obria la web app la base de dades s'esborrava i es començaven a afegir familiars des de zero.

Aquests errors es van solucionar i en aquest moment va ser en el que es va deixar provar la web app a persones totalment independents al projecte.

Aquesta va ser una tasca important ja que aquestes persones van ser capaces de donar urgències o detectar errors en els que no s'havia pensat abans.

Per últim, es van solucionar aquest errors i es va donar la web app per finalitzada.

7. Planificació i costos

7.1. Planificació temporal

Abans de començar a treballar en qualsevol projecte, cal realitzar una planificació temporal d'aquest. Per tant, abans de començar a dissenyar i programar la web app FamiSalut, es va dur a terme una planificació detallada de totes les tasques que calien realitzar-se i quant temps caldria per cadascuna.

A continuació es poden veure les fases del projecte i les tasques corresponents a cadascuna d'elles.

1. Definició del problema i recerca de solucions.
 - a. Definició del problema: el primer que es va fer va ser definir el problema i realitzar una pluja d'idees per les possibles solucions.
 - b. Recerca sobre tecnologies existents per solucionar el problema plantejat: un cop estudiades les possibles solucions al problema plantejat es va dur a terme un estudi sobre les possibles tecnologies amb les que dur a terme el projecte i es va escollir dissenyar i programar una aplicació web.
2. Disseny de la web app
 - a. Definició de les funcionalitats de la web app: en aquesta fase es va dur a terme un llistat de totes les funcionalitats que es volia que tingués la web app FamiSalut.
 - b. Disseny de la interfície: un cop especificades les funcionalitats que tindria FamiSalut es van dissenyar totes les pantalles que farien possibles que aquestes es complissin d'una manera senzilla i fàcil d'utilitzar.
 - c. Disseny de la base de dades: en aquesta fase, també es va determinar com es tractarien les dades necessàries pel funcionament de la web app.
3. Implementació
 - a. Estudi dels llenguatges de programació: abans de començar a programar la web app es va fer un estudi extens dels llenguatges amb els quals es programaria ja que els llenguatges HTML i CSS no eren llenguatges amb els que s'havia treballat freqüentment. Es van dur a terme proves i prototips de la web app abans de començar a programar la definitiva.
 - b. Programació de la web app: es va programar la interfície gràfica amb HTML i CSS i se li va donar una funcionalitat amb Python.

- c. Proves i detecció d'errors: al llarg de la programació de la web app s'han realitzat proves i correccions de la web app FamiSalut.
- 4. Redacció de la memòria
 - a. Per últim, s'ha redactat una memòria especificant els detalls més rellevants de l'elaboració del projecte.

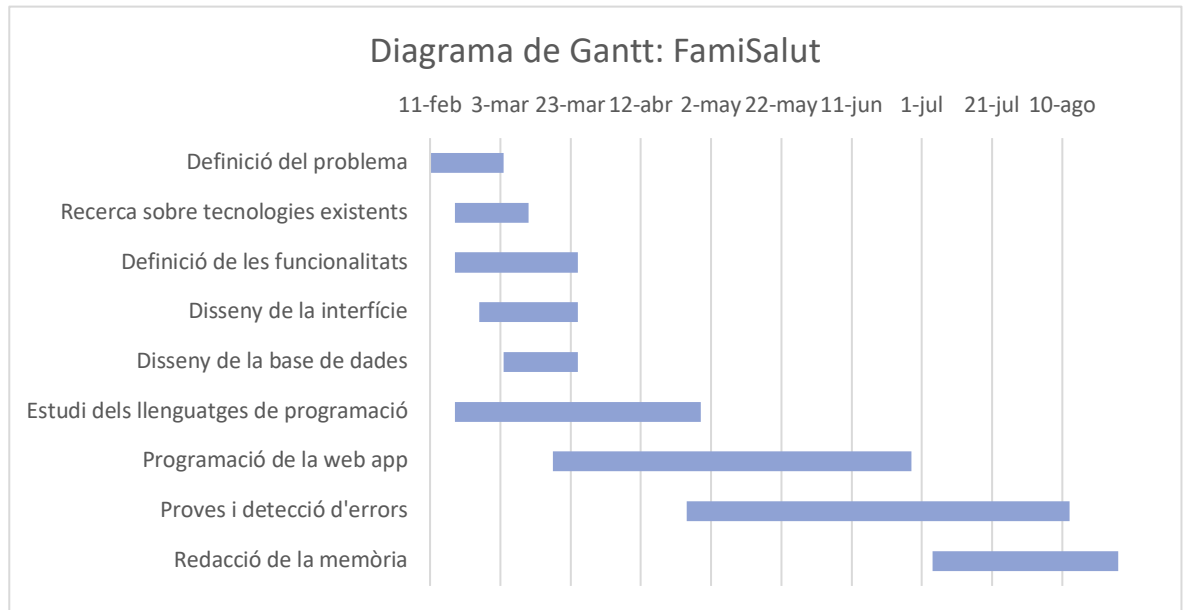
7.1.1. Duració de les fases

Un cop definides les fases del projecte, cal explicar quina ha sigut la distribució d'aquestes tasques al llarg del temps que ha durat el projecte.

A la Taula 1, es pot veure quant temps s'ha dedicat a cada fase i quin ha sigut el temps que s'ha trigat en desenvolupar el projecte i a la Taula 2 es pot veure un diagrama de Gantt.

Tasca	Temps emprat
Definició del problema i recerca de les solucions	25
Definició del problema	10
Recerca sobre tecnologies existents	15
Disseny de la web app	35
Definició de les funcionalitats	10
Disseny de la interfície	15
Disseny de la base de dades	10
Implementació	210
Estudi dels llenguatges de programació	50
Programació de la web app	130
Proves i detecció d'errors	30
Redacció de la memòria	40
Hores total invertides	310

Taula 1: Duració de les fases del projecte



Taula 2: Diagrama de Gantt

7.2. Costos

A continuació, es farà un estudi dels costos associats al projecte tenint en compte els següents elements:

- Salari del treballador: en aquest cas el d'un enginyer junior (35€/h). Per tant, si el projecte ha tingut una llargada de 310h els costos deguts al salari de l'enginyer son de 10.850€.

$$\text{Salari treballador} = 35 * 310 = 10.850$$

- Material: per a realitzar el projecte s'ha utilitzat un ordinador d'un cost de 1200€. S'ha estimat que s'ha treballat a mitja jornada durant 4 mesos per tal d'assolir les 340h de treball i es considera que l'ordinador s'amortitza passats els 2 anys, per tant, tenint en compte que només s'ha utilitzat durant 4 mesos el cost de material és de 200€.

$$\text{Cost material} = \frac{\text{Cost ordinador}}{24 \text{ mesos}} * \text{mesos utilitzat}$$

$$\text{Cost material} = \frac{1200}{24} * 4 = 200$$

Per tant, el cost final del projecte és de 11.050€. En la taula següent es pot veure un resum de la distribució de costos del projecte.

Concepte	Cost
Salari del treballador	10.850€
Material	200€
COST TOTAL	11.050€

Taula 3: Costos totals del projecte

6. Treball futur

L'objectiu d'aquest projecte ha sigut la realització d'un prototipus d'aplicació web que compleixi unes funcionalitats específiques. Degut als coneixements amb els quals s'ha iniciat el projecte i al temps programat per realitzar-lo, s'han escollit unes funcionalitats determinades deixant enrere algunes funcionalitats que també podien ser interessants.

A continuació s'exposaran algunes d'aquestes funcionalitats:

- Capacitat de penjar arxius a la web app per poder emmagatzemar receptes mèdiques, anàlisis o documents mèdics importants.
- Cercar per data per poder trobar una cita mèdica específica.
- Fer un apartat específic per emmagatzemar les receptes mèdiques de tots els familiars.

Per altra banda, s'ha de tenir en compte que no s'ha programat una base de dades sinó una estructura de dades amb l'ajut de fitxers .txt. Com a treball futur s'hauria de programar una base de dades per poder emmagatzemar més informació d'una manera més adequada. Tenint en compte la importància de la seguretat amb la que s'ha de comptar a l'hora d'emmagatzemar dades mèdiques ja que es tracta d'una informació altament sensible.

Es considera però, que el prototipus programat es una base vàlida per continuar treballant ja que s'han aconseguit programar totes les funcionalitats bàsiques i aquestes han donat un bon resultat a l'hora de fer servir l'aplicació web.

Conclusions

Un cop finalitzat aquest projecte cal fer un estudi del resultat que s'ha obtingut i de com s'ha arribat a aquest.

S'ha treballat al llarg d'un quadrimestre per convertir en realitat una idea inicial passant per totes les fases de desenvolupament d'un projecte (definició del problema, disseny de la solució i implementació d'aquesta).

Com resultat s'ha obtingut la web app FamiSalut. Una aplicació web que permet el seguiment de cites mèdiques de tota la família i que emmagatzema la informació mèdica més important de tots els familiars i metges que aquests visiten. Però a més a més, és una web app creada per que qualsevol persona la pugui utilitzar, des de qualsevol dispositiu i d'una manera fàcil i intuïtiva. Per tant, a l'utilitzar aquesta web app, els usuaris guanyaran temps i tranquil·litat a l'hora d'emmagatzemar la informació mèdica més important i a l'hora de realitzar un seguiment de totes les seves cites mèdiques i la dels seus familiars.

Per dur a terme la implementació d'aquesta web app s'ha utilitzat el framework Flask escrit amb Python i s'han creat documents HTML i CSS per la correcta visualització de la web app. Això ha implicat un estudi d'aquestes eines i una inversió en temps d'aprenentatge ja que no s'havia treballat amb elles prèviament. Per tant la implementació de la idea proposada a l'inici del projecte ha sigut la part on s'ha dedicat més temps ja que ha sigut la més complicada.

Per últim, cal mencionar que s'han assolit tots els objectius plantejats a l'inici del projecte tot i que el prototipus creat encara té espai de millora. S'han implementat les funcionalitats més bàsiques i importants però encara hi ha funcionalitats molt interessants que es podrien afegir en un futur.

Agraïments

M'agradaria dedicar un moment per donar les gràcies a les persones que m'han ajudat i que han contribuït a que aquest projecte fos possible.

Primer de tot vull agrair el suport que he rebut per part del meu tutor, Lluís Solano, ja que des del primer moment ha cregut en el projecte i m'ha animat a que el dugués a terme donant-me consells, dedicant-me tot el temps necessari i tenint en compte en tot moment que a l'hora de realitzar aquest projecte també estava cursant diverses assignatures.

Per altra banda, m'agradaria donar-li les gràcies a la meva mare no només per haver-me donat molts bons consells de com redactar la memòria sinó perquè va ser amb ella amb la que vaig tenir la idea inicial del projecte.

Bibliografia

Referències bibliogràfiques

- [1] REFSNES DATA, *W3Schools*, 1998, [<https://www.w3schools.com>, Febrer, Març, Abril, Maig, Juny i Juliol 2019]
- [2] DAVE SHEA; *CSS Zen Garden*, 2001, [<http://www.csszengarden.com/>, Març 2019]
- [3] CENTRO AYUDA GOOGLE, *Añadir un calendario de Google Calendar a tu sitio web*, 2019, [<https://support.google.com/calendar/answer/41207?hl=es>, Juny 2019]
- [4] GOOGLE PLAY, *Clepio - Gestiona tu salud*, 2019, [<https://play.google.com/store/apps/details?id=com.clepio&hl=es>, Juliol 2019]
- [5] GENERALIAT DE CATALUNYA, *La meva salut*, 2016, [<https://lamevasalut.gencat.cat/web/cps/preguntes-frequents>, Juliol 2019]
- [6] STACK OVERFLOW, *Can a html button perform a POST request?*, 2019, [<https://stackoverflow.com/questions/16036041/can-a-html-button-perform-a-post-request>, Juny 2019]
- [7] TOMMY OLSSON, *Posicionamiento absoluto y fijo con CSS*, Setembre 2016, [<https://mosaic.uoc.edu/ac/le/es/m6/ud11/index.html>, Abril 2019]
- [8] TRAVERSY MEDIA, *Responsive HTML & CSS Side Menu From Scratch*, Juny 2017, [<https://www.youtube.com/watch?v=wpGNFGqNfdU>, Març 2019]
- [9] JOSÉ DOMINGO, *¿Qué es Flask?*, Novembre 2017, [<https://openwebinars.net/blog/que-es-flask/>, Juliol 2019]

Bibliografia complementària

Al llarg del projecte s'han realitzat tres cursos MOOC de edX:

- [10] W3Cx, *CSS Basics: Learn how to take your Web pages from bland to bold with the power of CSS styling*, [<https://www.edx.org/es/course/css-basics>, Març 2019]
- [11] W3Cx, *HTML5 Coding Essentials and Best Practices*, [<https://www.edx.org/course/html5-coding-essentials-and-best-practices-2>, Març 2019]
- [12] UAMx, *Introducción al desarrollo de aplicaciones web*, [<https://www.edx.org/es/course/introduccion-al-desarrollo-de-aplicaciones-web-3>, Abril 2019]